

BAB II TINJAUAN PUSTAKA

2.1 *Support Vector Machine* (SVM)

Permodelan data empiris dapat menimbulkan beberapa permasalahan ketika data yang diperoleh berdimensi tinggi (ruang fitur) dan tidak seragam yang bisa mengakibatkan analisis dengan pendekatan *Neural Network* (NN) tradisional mengalami kesulitan dalam generalisasi dan menghasilkan model yang bisa *overfit* data. SVM dikembangkan untuk memecahkan masalah klasifikasi karena SVM memiliki kemampuan yang lebih baik dalam menggeneralisasi data bila dibandingkan dengan teknik yang sudah ada sebelumnya.

SVM merupakan sistem pembelajaran menggunakan ruang berupa fungsi-fungsi linear dalam sebuah ruang fitur berdimensi tinggi yang dilatih menggunakan algoritma pembelajaran berdasarkan pada teori optimasi dengan mengimplementasikan *learning bias*. Pendekatan dengan menggunakan SVM ini memiliki banyak manfaat lain seperti misalnya model yang dibangun memiliki ketergantungan eksplisit pada subset dari datapoints, serta *support vector* yang membantu dalam interpretasi model. Prinsip utama penggunaan SVM adalah mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada ruang input. *Hyperplane* tersebut berupa *line* pada *two dimension* dan dapat berupa *flat plane* pada *multiple plane*. SVM merupakan salah satu *machine learning* yang melakukan pelatihan dengan menggunakan *training dataset* dan melakukan generalisasi dan membuat prediksi dari data baru.

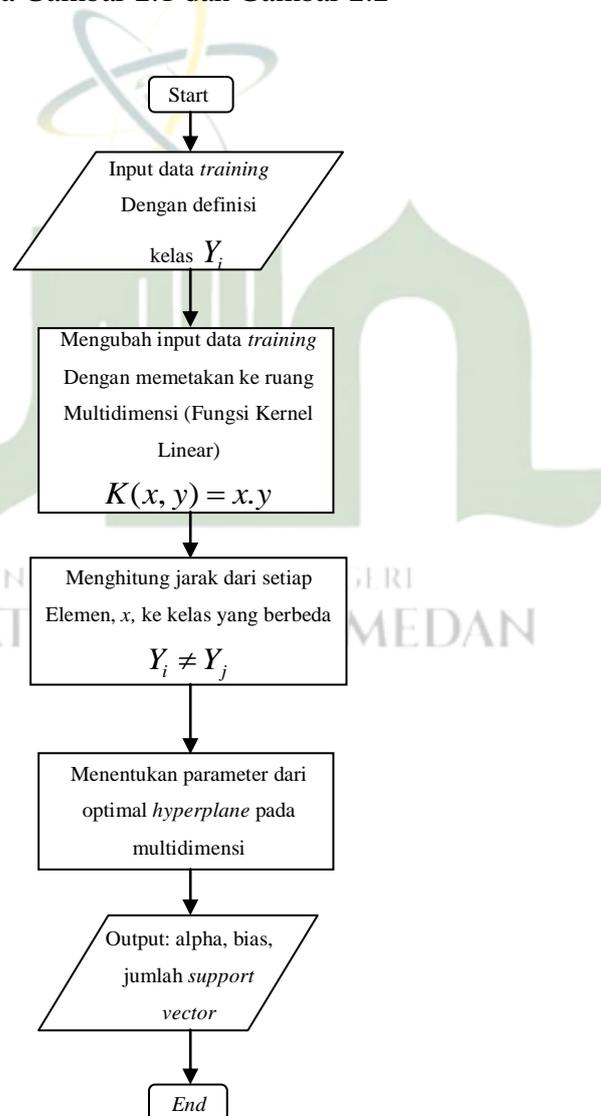
2.1.1 Konsep SVM

Support Vector Machine (SVM) merupakan metode klasifikasi jenis terpandu (*supervised*) karena ketika proses pelatihan, diperlukan target pembelajaran tertentu. SVM muncul pertama kali pada tahun 1992 oleh Vladimir Vapnik bersama rekannya Bernhard Boser dan Isabelle Guyon.

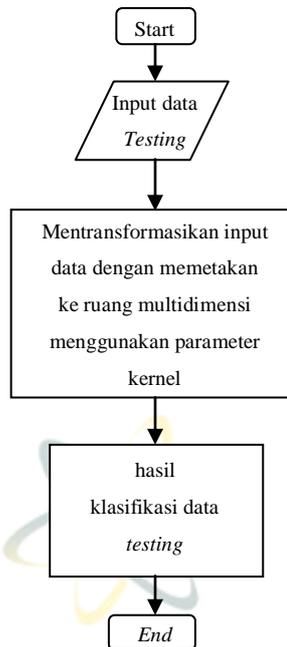
Ide dasar SVM adalah memaksimalkan batas *hyperplane* (*maximal margin hyperplane*). Konsep klasifikasi dengan SVM dapat dijelaskan secara sederhana

sebagai usaha untuk mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua kelas data pada ruan input. *Hyperplane* (batas keputusan) pemisah terbaik antara kedua kelas dapat ditemukan dengan menukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan data terdekat dari masing-masing kelas. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pelatihan pada SVM.

Dalam implementasinya, proses klasifikasi menjadi 2 tahap yaitu *training* dan *testing*. Diaram proses *training* pada SVM dapat dilihat pada Gambar dan diagram proses *testing* pada Gambar 2.1 dan Gambar 2.2



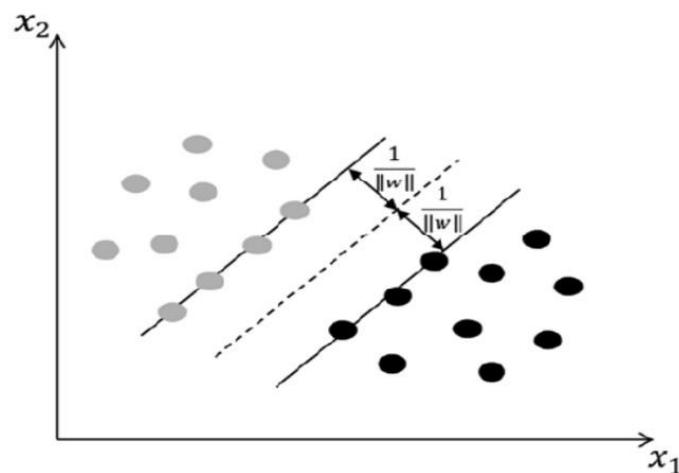
Gambar 2.1 Diagram Alur *Training SVM*



Gambar 2.2 Diagram Alur *Testing* SVM

2.1.2 *Hard – Margin* SVM / Linear SVM

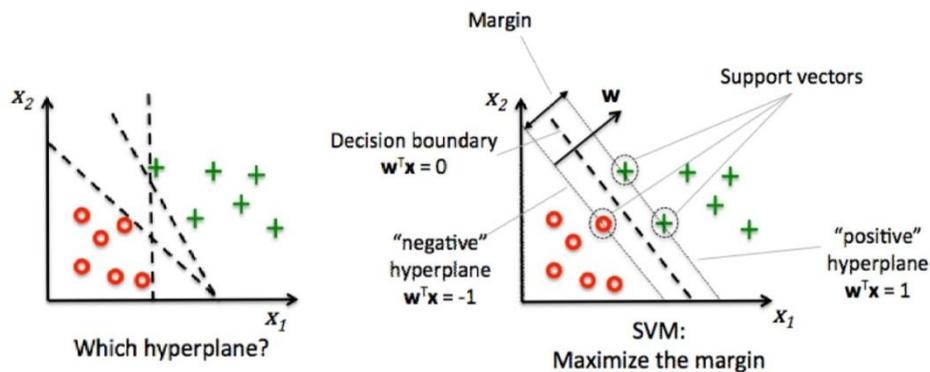
Teknik SVM merupakan klasifer yang menemukan *hyperplane* dengan kasus data yang digunakan merupakan data dengan dua kelas yang sudah terpisah secara linear seperti pada Gambar 2.3.



Sumber: Buku *Efficient Learning Machine*, 2015

Gambar 2.3 *Hard Margin* SVM

Berdasarkan pada Gambar 2.3, terlihat bahwa antara kelas positif dan kelas negatif sudah terpisah secara total terlihat dari lingkaran abu-abu yang berada dekat dengan garis x_2 sedangkan untuk lingkaran hitam terletak dekat dengan garis x_1 (Awad & Khanna, 2015).



Sumber: www.quora.com

Gambar 2.4 *Hyperplane* terbaik yang memisahkan antar dua kelas positif (+1) dan negatif (-1)

Berdasarkan Gambar 2.4 terlihat beberapa pola yang merupakan anggota dari dua buah kelas yaitu positif (+1) dan negatif (-1). *Hyperplane* terbaik dapat ditemukan dengan mengukur *margin hyperplane* dan mencari titik maksimalnya. *Margin* merupakan jarak antara *hyperplane* dengan data terdekat dari masing-masing kelas. Subset *training dataset* yang paling dekat dinamakan sebagai *support vector*. Pada Gambar 2.4 sebelah kanan menunjukkan *hyperplane* terbaik, yaitu terletak pada garis putus-putus yang berada tepat ditengah-tengah *hyperplane* positif dan *hyperplane* negatif. Sedangkan tanda “positif” dan “bulat” yang berada dalam lingkaran hitam merupakan *support vector*.

Pencarian lokasi *hyperplane* optimal merupakan inti dari metode SVM. Diasumsikan bahwa terdapat data *learning* dengan data *points* $x_i (i=1,2,\dots,m)$ memiliki dua kelas $y_i = \pm 1$ yaitu kelas positif (+1) dan kelas negatif (-1) sehingga akan diperoleh *decision function* berikut.

$$f(x) = \text{sign}(w \cdot x + b) \quad (2.1)$$

Dimana (\cdot) merupakan skalar sehingga $w \cdot x \equiv w^T x$

Berdasarkan pada *decision function* diatas, dapat terlihat bahwa data akan terklasifikasi secara tepat jika $y_i(w.x_i + b) > 0 \forall_i$ karena ketika $(w.x_i + b)$ harus bernilai positif saat $y_i = +1$, dan bernilai negatif ketika $y_i = -1$. *decision function* menjadi invarian ketika akan dilakukan pembuatan skala positif baru dari argumen dalam persamaan fungsi sehingga akan mengakibatkan ambiguitas dalam mendefinisikan konsep jarak atau margin. Maka dari itu didefinisikan skala untuk (w, b) dengan menetapkan $w.x + b = 1$ untuk titik terdekat pada satu sisi dan $w.x + b = -1$ untuk titik terdekat pada sisi lainnya. *Hyperplane* yang melewati $w.x + b = 1$ dan $w.x + b = -1$ disebut sebagai *hyperplane* kanonik dan wilayah antar *hyperplane* disebut sebagai *margin band*.

Margin maksimum dapat diperoleh dengan cara memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya yaitu $\frac{1}{\|w\|}$. Hal tersebut dirumuskan sebagai *Quadratic Programming (QP) Problem* dengan mencari titik minimal seperti pada persamaan berikut.

$$\min \tau(w) = \frac{1}{2} \|w\|^2 \quad (2.2)$$

Sedangkan subjek *constrain*/kendala persamaannya adalah sebagai berikut.

$$y_i(w.x_i + b) \geq 1 \forall_i \quad (2.3)$$

Persamaan diatas merupakan permasalahan optimisasi kendala dimana untuk meminimalkan fungsi objek pada persamaan (2) dengan kendala pada persamaan (3). Permasalahan diatas dapat direduksi dengan menggunakan fungsi *lagrange* yang terdiri dari jumlahan fungsi objektif dan m kendala dikalikan dengan pengganda *lagrange* seperti berikut.

$$L(w, b) = \frac{1}{2}(w.w) - \sum_{i=1}^m \alpha_i (y_i(w.x_i + b) - 1) \quad (2.4)$$

Dimana α_i merupakan *Lagrange Multipliers*, dan nilai $\alpha_i \geq 0$. Pada saat minimum, akan dilakukan penurunan dari b dan w dan mengaturny menjadi nol seperti berikut.

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^m \alpha_i y_i = 0 \quad (2.5)$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^m \alpha_i y_i x_i = 0 \quad (2.6)$$

Substitusi nilai w dari persamaan (6) kedalam bentuk $L(w,b)$ sehingga akan diperoleh rumus ganda atau biasa disebut sebagai *wofle dual*.

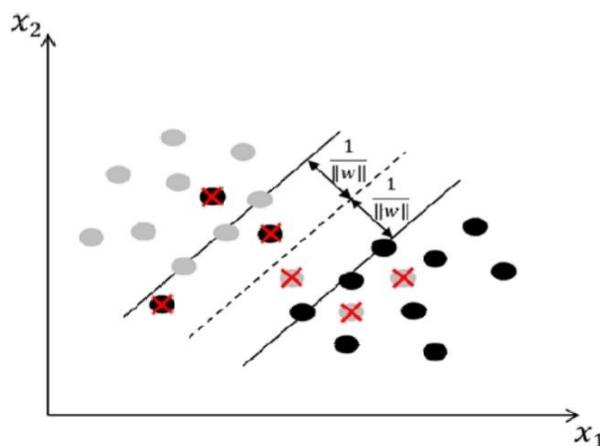
$$W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (2.7)$$

Dimana nilai α_i terhadap kendala adalah sebagai berikut.

$$\alpha_i \geq 0 \quad \sum_{i=1}^m \alpha_i y_i = 0 \quad (2.8)$$

2.1.3 Soft-Margin SVM

Ketika data yang digunakan tidak sepenuhnya dapat dipisahkan, *slack variables* x_i diperkenalkan kedalam fungsi obyektif SVM untuk memungkinkan kesalahan dalam misklasifikasi. Dalam hal ini, SVM bukan lagi *hard margin classifier* yang akan mengklasifikasi semua data dengan sempurna melainkan sebaliknya yaitu SVM *soft margin classifier* dengan mengklasifikasikan sebagian besar data dengan benar, sementara memungkinkan model untuk membuat misklasifikasi beberapa titik disekitar batas pemisah. Berikut merupakan gambar ketika data termasuk kedalam *soft margin SVM* (Awad & Khanna, 2015).



Sumber: Buku *Efficient Learning Machine*, 2015

Gambar 2.5 Misklasifikasi pada *soft margin SVM*

Berdasarkan pada Gambar 2.5 diatas, terlihat bahwa data pada kedua kelas tidak terpisah secara sempurna dapat dilihat dari beberapa lingkaran abu-abu yang persebarannya berada di sekitar area lingkaran hitam serta sebaliknya terdapat beberapa lingkaran hitam yang persebarannya berada di sekitar lingkaran abu-abu. Persamaan *soft margin* hampir mirip dengan *hard margin* hanya terdapat sedikit modifikasi dengan adanya *slack variabel* pada persamaan (3) sebelumnya seperti berikut ini.

$$y_i(w \cdot x_i + b) \geq 1 - \varepsilon_i \quad (2.9)$$

Kemudian ketika akan dilakukan minimasi jumlahan eror $\sum_{i=1}^m \xi_i$ adalah sebagai berikut.

$$\min \left[\frac{1}{2} w \cdot w + C \sum_{i=1}^m \varepsilon_i \right] - \sum_{i=1}^m \alpha_i [y_i(w \cdot x_i + b) - 1 + \varepsilon_i] - \sum_{i=1}^m r_i \varepsilon_i \quad (2.10)$$

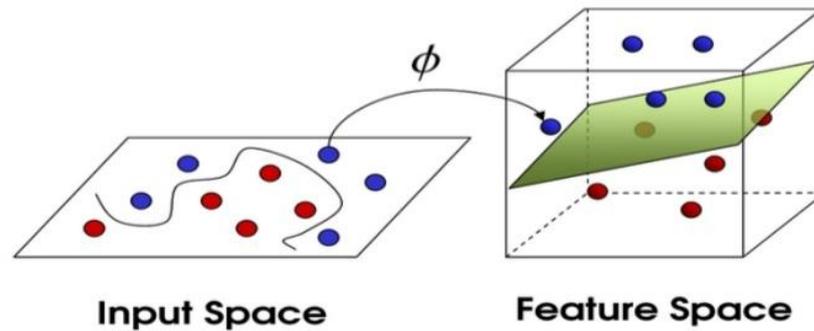
Dengan demikian, persamaan (2) akan diubah kedalam persamaan berikut.

$$\min \tau(w) = \frac{1}{2} \|w\|^2 + c \sum_{i=1}^m \varepsilon_i \quad (2.11)$$

Parameter c digunakan untuk mengontrol *trade off* antara *margin* dan kesalahan klasifikasi ε (Abthoni,2017).

2.1.4 Kernel SVM

Ketika terdapat permasalahan data yang tidak terpisah secara linear dalam ruang input, *soft margin SVM* tidak dapat menemukan *hyperplane* pemisah yang kuat untuk meminimalkan misklasifikasi dari *data points* serta menggeneralisasi dengan baik. Untuk itu, kernel dapat digunakan untuk mentransformasi data ke ruang berdimensi lebih tinggi yang disebut sebagai ruang kernel. Dimana akan menjadikan data terpisah secara linear (Awad & Khanna,2015).



Sumber: www.quora.com

Gambar 2.6 Kernel SVM untuk memisahkan data secara linear

Data disimpan dalam bentuk kernel yang mengukur kesamaan atau ketidaksamaan objek data. Kernel dapat dibangun untuk berbagai objek data mulai dari data kontinu dan data diskrit melalui urutan data dan grafik. Konsep substitusi kernel berlaku bagi metode lain dalam analisis data. Tetapi SVM merupakan yang paling terkenal dari metode dengan jangkauan kelas luas yang menggunakan kernel untuk mempresentasikan data dan dapat disebut sebagai metode berbasis kernel. Berikut merupakan ilustrasi contoh dalam melakukan pemisahan data menggunakan kernel. Diketahui bahwa data terdiri dari *input space* dengan dua buah $\mathbf{x} = \{x_1, x_2\}$ dan $\mathbf{z} = \{z_1, z_2\}$. Diasumsikan fungsi kernel akan dibuat dengan menggunakan input x dan z seperti berikut.

$$\begin{aligned}
 \mathbf{K}(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^T \mathbf{z})^2 \\
 \mathbf{K}(\mathbf{x}, \mathbf{z}) &= (x_1 z_1 + x_2 z_2)^2 \\
 \mathbf{K}(\mathbf{x}, \mathbf{z}) &= (x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2)^2 \\
 \mathbf{K}(\mathbf{x}, \mathbf{z}) &= (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^T (z_1^2, \sqrt{2}z_1 z_2, z_2^2) \\
 \mathbf{K}(\mathbf{x}, \mathbf{z}) &= \Phi(x)^T \Phi(z)
 \end{aligned} \tag{2.12}$$

Nilai \mathbf{K} diatas secara implisit mendefinisikan pemetaan ke ruang dimensi yang lebih tinggi seperti berikut.

$$\Phi(x) = \{x_1^2, \sqrt{2}x_1 x_2, x_2^2\} \tag{2.13}$$

Kernel $\mathbf{K}(\mathbf{x}, \mathbf{z})$ mengambil dua *input space* dan memberikan kesamaannya dalam *feature space* seperti berikut.

$$\begin{aligned}\Phi: X &\rightarrow F \\ \mathbf{K}: X \times X &\rightarrow R, \mathbf{K}(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{z})\end{aligned}\quad (2.14)$$

Berdasarkan pada fungsi kernel diatas, dapat dilakukan perhitungan untuk melakukan prediksi dari beberapa data dalam *feature space* seperti pada persamaan berikut.

$$\begin{aligned}f(\Phi(x)) &= \text{sign}(w \cdot \Phi(z) + b) \\ f(\Phi(x)) &= \text{sign}\left(\sum_{i=1}^m \alpha_i y_i \mathbf{K}(\mathbf{x}, \mathbf{z}) + b\right)\end{aligned}\quad (2.15)$$

Dimana :

b : nilai bias

m : jumlah *support vector*

$\mathbf{K}(\mathbf{x}, \mathbf{z})$: fungsi kernel

Nilai k yang bisa digunakan sebagai fungsi kernel harus memenuhi kondisi Mercer antara lain:

1. Merupakan *Hilbert Space* dimana nilai *feature space* harus merupakan vektor dengan *dot product*.
2. Harus benar jika k merupakan fungsi definit positif

$$\int dx \int dz f(x) \mathbf{K}(\mathbf{x}, \mathbf{z}) f(z) > 0 \quad (\forall f \in L_2) \quad (2.16)$$

3. Ketika k_1 dan k_2 merupakan fungsi kernel, maka :

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \mathbf{K}_1(\mathbf{x}, \mathbf{z}) + \mathbf{K}_2(\mathbf{x}, \mathbf{z}) : \text{Direct sum} \quad (2.17)$$

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \alpha \mathbf{K}_1(\mathbf{x}, \mathbf{z}) : \text{Skalar product} \quad (2.18)$$

$$\mathbf{K}(\mathbf{x}, \mathbf{z}) = \mathbf{K}_1(\mathbf{x}, \mathbf{z}) \mathbf{K}_2(\mathbf{x}, \mathbf{z}) : \text{Direct product} \quad (2.19)$$

Berikut merupakan fungsi kernel yang populer dan sering digunakan antara lain sebagai berikut.

1. Linear Kernel SVM

Linear kernel merupakan fungsi kernel yang paling sederhana. Linear kernel digunakan ketika data yang dianalisis sudah terpisah secara linear. Linear kernel cocok ketika terdapat banyak fitur dikarenakan pemetaan ke ruang dimensi yang lebih tinggi tidak benar-benar meningkatkan kinerja seperti pada klasifikasi teks. Dalam klasifikasi teks, baik jumlah *instances* (dokumen) maupun jumlah

fitur (kata) sama-sama besar (Kowalczyk,2014). Berikut merupakan dari linear kernel SVM.

$$K(x, y) = x.y \quad (2.20)$$

Pemetaan fungsi Φ merupakan identitas / tidak ada pemetaan.

2. *Polynomial Kernel*

Polinomial kernel merupakan fungsi kernel yang digunakan ketika data tidak terpisah secara linear. Polinomial kernel sangat cocok untuk permasalahan dimana semua *training dataset* dinormalisasi.

$$K(x, y) = (x.y + 1)^p \quad (2.21)$$

3. *Radial Basis Function (RBF) Kernel*

RBF kernel merupakan fungsi kernel yang biasa digunakan dalam analisis ketika data tidak terpisah secara linear. RBF kernel memiliki dua parameter yaitu *Gamma* dan *Cost*. Parameter *cost* atau biasa disebut sebagai *C* merupakan parameter yang bekerja sebagai pengoptimalan SVM untuk menghindari misklasifikasi di setiap sampel dalam *training dataset*. Parameter *gamma* menentukan seberapa jauh pengaruh dari satu sampel *training dataset* dengan nilai rendah yang berarti “jauh”, dan nilai tinggi berarti “dekat”. Dengan *gamma* yang rendah, titik yang berada jauh dari garis pemisah yang masuk akan dipertimbangkan dalam perhitungan untuk garis pemisah. Ketika *gamma* tinggi berarti titik-titik berada di sekitar garis yang masuk akan dipertimbangkan dalam perhitungan. Berikut merupakan persamaan dari RBF kernel.

$$\mathbf{K}(x, y) = \exp\left[-\gamma\|x - y\|^2\right] \quad (2.22)$$

2.2 *Matriks Hessian*

Matriks *Hessian* merupakan suatu elemen yang dibentuk dari turunan parsial kedua dari suatu fungsi. Pada metode SVM, matriks *hessian* dihasilkan dari perhitungan kernel SVM dengan nilai kelas y_i . Fungsi dari matriks *hessian* adalah untuk mengidentifikasi optimum relatif suatu nilai fungsi. Untuk menghitung matriks *hessian*, dapat ditunjukkan pada Persamaan 2.23.

$$[D]_{ij} = y_i y_j (K(x, y) + \lambda^2) \quad (2.23)$$

Keterangan :

$[D]_{ij}$ = nilai matriks *hessian* ke- i, j

$K(x, y)$ = hasil perhitungan kernel

λ^2 = nilai lambda yang telah diinisialisasi

2.3 Sequential Training Support Vector Machine

Sequential Training Support Vector Machine merupakan algoritma dalam proses *training* dari metode *training*, *sequential training support vector machine* adalah yang paling sederhana. Selain itu model *training* ini lebih mudah dimengerti dan dapat meminimalkan waktu. Langkah-langkah dalam *sequential training support vector machine* adalah :

1. Menginisialisasi nilai parameter seperti nilai *Complexity* (C), λ , γ , α , ε
2. Mencari data dari i sampai j , kemudian mencari nilai matriks *hessian* berdasarkan Persamaan 2.23.
3. Mengulangi tahap ketiga sampai nilai iterasi yang didapatkan sampai pada iterasi maksimum. Pada tahap ketiga, pertama mencari nilai E_i yang dapat ditunjukkan pada Persamaan 2.24. kemudian menghitung $\delta\alpha_i$ yang ditunjukkan oleh Persamaan 2.25. Lalu, menghitung nilai α_i yang dapat ditunjukkan pada Persamaan 2.26.

$$E_i = \sum_j^n 1\alpha_i D_{ij} \quad (2.24)$$

$$\delta\alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \quad (2.25)$$

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (2.26)$$

Keterangan :

α_i = nilai parameter *alpha*

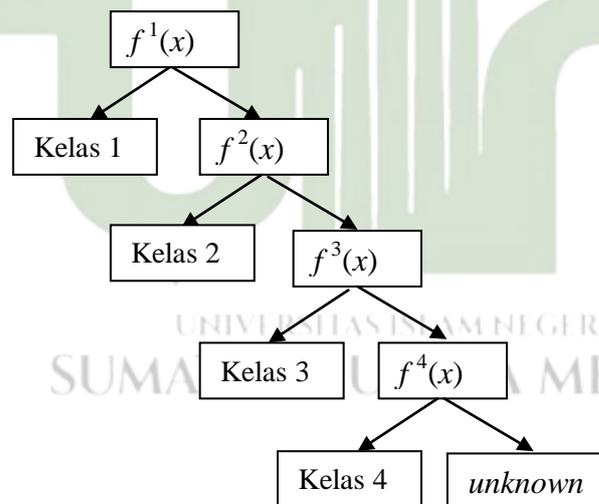
D_{ij} = hasil perhitungan matriks *hessian*

C = nilai parameter *complexity*

$\delta\alpha_i$ = parameter *delta alpha* data ke- i

2.4 *One Against All*

One against all merupakan suatu metode yang digunakan dalam SVM untuk memecahkan masalah *multi class* atau hasil klasifikasi yang memiliki lebih dari 2 kelas. Untuk mengimplementasikan *multi class* dalam SVM adalah dengan menggabungkan beberapa SVM *biner* ataupun semua data yang terdiri dari beberapa kelas kedalam bentuk permasalahan optimasi. Dengan menggunakan metode *one against all* maka harus dibangun k buah model SVM *biner* (k merupakan jumlah kelas). Setiap model klasifikasi ke- i dilatih dengan menggunakan keseluruhan data, untuk mencari solusi permasalahan baik label kelas positif maupun label kelas negatif pada seluruh sampel lainnya. Misalnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Untuk pelatihan digunakan 4 buah SVM *biner* seperti pada Gambar 2.7.



Gambar 2.7 Klasifikasi Menggunakan *One Against All* Untuk 4 Kelas

2.5 *Confusion Matrix*

Confusion Matrix atau *error matrix* merupakan sebuah matriks yang menampilkan visualisasi kinerja dari algoritma klasifikasi menggunakan data dalam matriks. Hal tersebut membandingkan klasifikasi prediksi terhadap klasifikasi aktual dalam bentuk *False Positif* (FP), *True Positif* (TP), *False*

Negatif (FN), dan *True Negatif* (TN) dari informasi. *Confusion Matrix* untuk sistem klasifier dua kelas adalah sebagai berikut.

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

Sumber: www.mathworks.com

Gambar 2.8 *Confusion Matrix*

Berdasarkan pada Gambar 2.7, dijelaskan bahwa ketika berada pada kolom TP maka hasil bernilai benar dan teridentifikasi sebagai positif. Ketika hasil berada pada kolom FP maka hasil adalah salah teridentifikasi sebagai positif. Ketika hasil berada pada kolom FN maka hasil adalah salah dan teridentifikasi sebagai negatif dan ketika hasil berada di kolom TN, maka akan bernilai benar dan teridentifikasi negatif.

2.6 Akurasi

Akurasi atau tingkat kesalahan merupakan angka prediksi yang benar (atau salah) yang dibuat oleh model melalui kumpulan dari data. Akurasi biasanya dihitung dengan menggunakan tes independen yang tidak selalu digunakan dalam proses pembelajaran. Teknik estimasi akurasi yang lebih kompleks seperti *bootstrapping* ataupun *cross validation* umumnya digunakan terutama untuk *dataset* yang memiliki sampel kecil (Awad & Khanna,2015).

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FN + FP} \quad (2.27)$$

2.7 Uji *Chi-Square*

Sesuai dengan namanya, statistik uji yang digunakan dalam metode SVM adalah statistik uji *chi-square*. statistik uji *chi-square* dapat digunakan untuk mengetahui independensi (kebebasan) antara dua variabel. Misalkan dua variabel akan diuji independensinya, yang mana variabel pertama mempunyai r kategori dan variabel kedua mempunyai c kategori. maka struktur data uji *chi-square* dapat dilihat pada tabel. 2.1.

Tabel 2.1 Struktur Data Uji *Chi-Square*

Faktor I	Faktor II					Jumlah
	B ₁	...	B _j	...	B _k	
A ₁	n ₁₁	...	n _{1j}	...	n _{1k}	n ₁
.
A _i	n _{i1}	...	n _{ij}	...	n _{ik}	n _i
.
A _b	n _{b1}	...	n _{bj}	...	n _{bk}	n _b
Jumlah	n ₁	...	n _j	...	n _k	N

Sumber: Nugraha(2016)

Keterangan

n_{11} : Banyaknya pengamatan dengan kategori A₁ dengan B₁

n_{ij} : Banyaknya pengamatan yang termasuk dalam kategori ke- i dari variabel pertama dan kategori ke- j dari variabel kedua

n_i : Banyaknya pengamatan yang termasuk dalam kategori ke- i

n_j : Banyaknya pengamatan yang termasuk dalam kategori ke- j

Hipotesis yang digunakan pada pengujian *chi-Square* adalah sebagai berikut:

H₀ : Kedua kriteria klasifikasi adalah saling bebas (tidak terdapat hubungan antara variabel pertama dan variabel kedua atau independen)

H_1 : kedua kriteria klasifikasi adalah tidak saling bebas (terdapat hubungan antara variabel pertama dan variabel kedua atau dependen)

Taraf signifikansi : α

Statistik Uji :

$$X^2 = \sum_{i=1}^b \sum_{j=1}^k \frac{(n_{ij} - E_{ij})^2}{E_{ij}} \quad (2.28)$$

Keterangan :

n_{ij} : Banyaknya pengamatan yang termasuk dalam kategori ke- i dari variabel pertama dan kategori ke- j dari variabel kedua

E_{ij} : Frekuensi harapan pengamatan yang termasuk dalam kategori ke- i dari variabel pertama dan kategori ke- j dari variabel kedua

b : Jumlah kategori dalam variabel pertama

k : Jumlah kategori dalam variabel kedua

Untuk menghitung frekuensi harapan masing-masing sel digunakan rumus:

$$E_{ij} = \frac{n_i \cdot n_j}{n} \quad (2.29)$$

Keterangan :

E_{ij} : Frekuensi harapan

n : Banyaknya seluruh pengamatan

n_i : Banyaknya pengamatan yang termasuk dalam kategori ke- i dari variabel pertama

n_j : Banyaknya pengamatan yang termasuk dalam kategori ke- j dari variabel kedua

kriteria pengambilan keputusan dalam uji *chi-square* yaitu H_0 ditolak jika atau dengan membandingkan nilai signifikansi dengan taraf signifikansi (α).

Statistik uji *chi-square* digunakan dalam dua cara dalam analisis SVM. Pertama, untuk menentukan apakah kategori-kategori dalam sebuah variabel independen bersifat seragam dan bisa digunakan menjadi satu. Kedua, ketika semua variabel independen sudah diringkas menjadi bentuk yang signifikan dan tidak mungkin digabung lagi, maka statistik uji *chi-square* digunakan untuk

menentukan variabel independen mana yang paling signifikan untuk membagi kategori-kategori dalam variabel dependen.

2.8 Uang Kuliah Tunggal (UKT)

Uang Kuliah Tunggal yang selanjutnya disingkat UKT merupakan sebagian dari biaya kuliah tunggal yang ditanggung oleh setiap mahasiswa pada setiap jurusan atau program studi untuk program diploma dan program sarjana. UKT merupakan besaran biaya yang harus dibayarkan oleh mahasiswa pada setiap semester. Universitas Islam Negeri Sumatera Utara menggunakan sistem UKT yang dikelompokkan berdasarkan kemampuan ekonomi mahasiswa, orang tua atau pihak yang membiayai. UKT ini dibayarkan setiap semester dengan jumlah yang sama sampai lulus pendidikan dan tidak ada biaya lagi selain UKT tersebut. UKT berlaku untuk semua mahasiswa jalur penerimaan baik reguler (SNMPTN & SPAN PTKIN) maupun jalur mandiri.

Adapun variabel-variabel sebagai pertimbangan dalam penentuan UKT antara lain :

1. Golongan UKT
2. Pendidikan Ayah
 - a. SD
 - b. Tidak Tamat SD
 - c. SLTP
 - d. SLTA
 - e. Sarjana
3. Pendidikan Ibu
 - a. SD
 - b. Tidak Tamat SD
 - c. SLTP
 - d. SLTA
 - e. Sarjana
4. Penghasilan Ayah
 - a. 500 – 900 Ribu
 - b. 1 – 1,9 Juta

- c. 2 – 2,9 Juta
 - d. 3 – 3,9 Juta
 - e. 4 - 4,9 Juta
 - f. 5 – 5,9 Juta
 - g. 6 - 7 Juta
 - h. Tidak Berpenghasilan
5. Penghasilan Ibu
- a. 500 – 900 Ribu
 - b. 1 – 1,9 Juta
 - c. 2 – 2,9 Juta
 - d. 3 – 3,9 Juta
 - e. 4 - 4,9 Juta
 - f. 5 – 5,9 Juta
 - g. 6 - 7 Juta
 - h. Tidak Berpenghasilan
6. Jumlah Tanggungan Orang Tua
- a. < 3 Orang
 - b. 3 – 5 Orang
 - c. 6 – 7 Orang
 - d. > 7 Orang

2.9 Penelitian Terdahulu (Relevan)

Berdasarkan penelitian yang telah dilakukan oleh Putra (2018), dengan judul penelitiannya adalah “Penerapan *Naïve Bayes Classifier* dengan *Gaussian Function* Untuk Menentukan Kelompok UKT”. Pada penelitian yang beliau lakukan adalah dari proses perancangan hingga pengujian yang telah dilakukan pada sistem pendukung keputusan dalam menentukan kelompok UKT mahasiswa, hasil validasi pengujian model klasifikasi NBC dengan *K-fold cross validation* menghasilkan rata-rata akurasi sebesar 86.67%, sehingga tingkat efektifitas model klasifikasi NBC dapat dikatakan cukup baik, sehingga sistem ini dapat digunakan

sebagai rekomendasi dalam pengambilan keputusan mengenai penentuan kelompok UKT mahasiswa baru yang diterima di IAIN Zawiyah Cot Kala Langsa.

Adapun penelitian yang telah dilakukan oleh Sutoyo dan Sumpala (2017), dengan judul penelitiannya adalah “*Case Based Reasoning* Menentukan Kelompok UKT (Studi Universitas Sembilan Belas November Kolaka)”. Pada penelitian yang telah dilakukan pencarian nilai *similarity* cukup dilakukan terhadap kasus yang memiliki indeks yang sama dengan kasus baru. Dari hasil pembahasan terdapat 41 kasus yang terindexing, dimana diperoleh kemiripan tertinggi dari hasil pengujian kasus baru yaitu 0.920 dan terdapat kasus ke-13. Dari hasil yang diperoleh antara kemiripan kasus baru terhadap kasus lama dapat dijadikan bahan pertimbangan (usulan) untuk menentukan kelompok UKT.

Pada penelitian yang dilakukan oleh Kurniawan *et al.* (2020), dengan judul penelitiannya ialah “*Data Mining* Menggunakan Metode *K-Means Clustering* Untuk Menentukan Besaran Uang Kuliah Tunggal”. Pada hasil penelitian yang telah dilakukan adalah dari proses *K-Means Clustering* pengelompokan besaran UKT calon mahasiswa baru terdiri dari 5 kelompok yaitu kelompok UKT kategori 1 dengan besaran UKT Rp.500.000,00, kelompok UKT kategori 2 dengan besaran UKT Rp.1.000.000,00, kelompok UKT kategori 3 dengan besaran UKT Rp.2.000.000,00, kelompok UKT kategori 4 dengan besaran UKT Rp.3.000.000,00, kelompok UKT kategori 5 dengan besaran UKT Rp.4.000.000,00. Penerapan *data mining* dengan menggunakan metode *K-Means Clustering* dapat membantu dalam menentukan besaran UKT calon mahasiswa baru.

Adapun penelitian yang telah dilakukan oleh Maslihah (2019), dengan judul penelitiannya yaitu “Aplikasi Logika *Fuzzy* Untuk Menentukan Uang Kuliah Tunggal (UKT)”. Pada penelitian yang beliau lakukan adalah berdasarkan sampel pengujian yang peneliti ambil sebanyak 96 orang mahasiswa baru tahun akademik 2018-2019 maka dana yang masuk dari hasil UKT sebanyak Rp.398.364.000 sedangkan dana yang masuk dari hasil UKT menggunakan ilmu logika *fuzzy* sebesar Rp.399.175.000. jadi terdapat selisih sebesar Rp.811.000.000 lebih tinggi logika *fuzzy*. Berdasarkan sampel data yang beliau ambil sebanyak 96

orang mahasiswa baru terdapat minimal 30% mahasiswa yang melakukan banding UKT. Hal ini menunjukkan adanya ketidakpuasan atas golongan yang telah ditetapkan dengan kondisi keuangan mahasiswa.

Pada penelitian yang telah dilakukan oleh Sahara (2017), dengan judul penelitiannya adalah “Penerapan Metode *Support Vector Machine* (SVM) Guna Menentukan Tingkat Lulus Mahasiswa E-Learning”. Dengan hasil penelitian yang telah beliau lakukan yaitu berdasarkan pengolahan data yang sudah dilakukan dengan membandingkan algoritma *support vector machine* dengan menggunakan data log mahasiswa. Kemudian di uji untuk mendapatkan nilai *accuracy* dan AUC dari setiap algoritma sehingga didapat hasil pengujian tertinggi dengan menggunakan *support vector* dengan nilai *accuracy* sebesar 81.02%, dan nilai AUC sebesar 0.810. dibandingkan dengan hanya menggunakan *support vector machine* (SVM) hasil ini lebih tinggi dengan selisih nilai akurasi sebesar 1.77% dan selisih AUC 0.006.

Adapun penelitian yang telah dilakukan oleh karim (2017) yaitu “Penentuan Besaran Uang Kuliah Tunggal untuk Mahasiswa Baru di Universitas Sam Ratulangi Menggunakan *Data Mining*”. Pada penelitian, data dianalisa dengan menggunakan algoritma C4.5 untuk mendapatkan model yang terbentuk sebagai pohon keputusan yang diolah menggunakan *tools RapidMiner*. Data penelitian yang diambil dari data kuisisioner yang dibagikan kepada mahasiswa Sam Ratulangi dengan kriteria mahasiswa angkatan 2013-2016 dan dilakukan dalam dua percobaan yaitu percobaan 1 dengan jumlah 80 kuisisioner diperoleh salah satu aturan yang ada pada pohon keputusan yaitu jika penghasilan ibu = Rp.4.000.000-Rp.5.000.000 maka kelas UKT-nya kelompok 2. Pada percobaan 2 dengan jumlah 115 kuisisioner diperoleh salah satu aturan yang ada pada pohon keputusan yaitu jika kondisi ibu = masih hidup dan pekerjaan ayah = tukang maka kelas UKT-nya kelompok 1.

2.10 Konsep Wahdatul Ulum

Islam merupakan agama yang memuliakan dan mewajibkan umatnya untuk menuntut ilmu. Islam sangat menekankan umatnya untuk selalu belajar, karena orang yang berilmu memiliki kedudukan yang lebih tinggi daripada orang yang tidak memiliki ilmu. Namun pada kenyataannya tidak semua orang bisa mendapatkan pendidikan dikarenakan faktor ekonomi yang tidak mendukung, belum lagi biaya pendidikan yang mahal khususnya pendidikan pada tingkat Perguruan Tinggi.

pada tahun 2013 pemerintah mulai menerapkan sistem pembayaran baru yang disebut dengan Uang Kuliah Tunggal (UKT). Sistem UKT ini menerapkan sistem subsidi silang, dimana mahasiswa dengan tingkat ekonomi tinggi akan mensubsidi mahasiswa dengan tingkat ekonomi rendah. hal tersebut mempunyai maksud yang selaras seperti kandungan isi surat Al-Dzariyat 51:19 :

وَفِي أَمْوَالِهِمْ حَقٌّ لِّلسَّائِلِ وَالْمَحْرُومِ

Artinya: “Dan pada harta-harta mereka ada hak untuk orang miskin yang meminta dan orang miskin yang tidak mendapat bagian”. (QS. Al-Dzariyat 51:19)

Dalam UKT mahasiswa akan dibagi menjadi 7 kelompok berdasarkan tingkat ekonominya. Hal yang berkaitan dengan pengelompokan juga disebutkan dalam Al-Qur’an Surat Al-Huud Ayat 24 yang berbunyi :

﴿مَثَلُ الْفَرِيقَيْنِ كَالْأَعْمَى وَالْأَصْمَى وَالْبَصِيرِ وَالسَّمِيعِ
هَلْ يَسْتَوِيَانِ مَثَلًا أَفَلَا نَذَكَّرُونَ﴾ (٢٤)

Artinya: “Perbandingan kedua golongan itu (orang-orang kafir dan orang-orang mukmin), seperti orang buta dan tuli dengan orang yang dapat melihat dan dapat mendengar. Adakah kedua golongan itu sama keadaan dan sifatnya?. Maka

tidaklah kamu mengambil pelajaran (daripada perbandingan itu)?.”(QS. Al-Huud Ayat 24)

Agar pengelompokan golongan UKT sesuai dengan keadaan ekonomi mahasiswa yang sebenarnya, maka dibutuhkanlah suatu sistem agar hasil pengelompokan dapat dirasa adil menurut keadaan ekonomi. Keadilan yang dimaksud ialah adil dalam hal kemampuan membayar uang kuliah. Berlaku adil sangat ditekankan dalam islam.

